UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/099,721	03/14/2002	Gregory E. James	NVIDP074/P000427	1906

28875 7590 12/06/2007

Zilka-Kotab, PC
P.O. BOX 721120
SAN JOSE, CA 95172-1120

EXAMINER	
GUILL, RUSSELL L	
ART UNIT	**PAPER NUMBER**
2123	
MAIL DATE	**DELIVERY MODE**
12/06/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No. 10/099,721	Applicant(s) JAMES, GREGORY E.
	Examiner Russ Guill	Art Unit 2123

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *12 October 2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,2,4,5,7-15,17,18 and 20-31* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,2,4,5,7-15,17,18 and 20-31* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *14 March 2002* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This non-final Office Action is in response to an Amendment filed October 12, 2007. No claims were added or cancelled. Claims 1 - 2, 4 - 5, 7 - 15, 17 - 18 and 20 – 31 are pending. Claims 1 - 2, 4 - 5, 7 - 15, 17 - 18 and 20 – 31 have been examined. Claims 1 - 2, 4 - 5, 7 - 15, 17 - 18 and 20 – 31 have been rejected.

2. **This Office Action is NON-final due to new rejections.**

Response to Remarks

3. Regarding all argued claims, the Applicant's arguments are partly persuasive, and partly unpersuasive, as discussed below.

a. The Applicant argues:

b. The Examiner has rejected Claims 1-2, 10-15, 17-18, 20-23, and 27 under 35 U.S.C. 103(a) as being unpatentable over Press et al. ("Numerical Recipes in Fortran 77"), in view of Rumpf et al. ("Using Graphics Cards for Quantized FEM Computations"), and in further view of Roy-Chowdhury ("Algorithm-Based Error-Detection Schemes for Iterative Solution of Partial Differential Equations"). Further, the Examiner has rejected Claims 26, 28, and 30-31 under 35 U.S.C. 103(a) as being unpatentable over Press et al. ("Numerical Recipes in C"), hereinafter Press2, in view of Rumpf, and in further view of Roy-Chowdhury. Moreover, the Examiner has rejected Claim 29 under 35 U.S.C. 103(a) as being unpatentable over Press2, in view of RoyChowdhury, and in further view of Rumpf. Applicant respectfully disagrees with such rejections.

c. To establish *a prima facie case* of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art and not based

on applicant's disclosure. *In re Vaeck,947* F.2d 488, 20 USPQ2d 1438 (Fed.Cir.1991).

d. With respect to the first element of the prima *facie case* of obviousness, the Examiner has stated that "the motivation to use the art of Rumpf with the art of Press would have been the benefits recited in Rumpf that the presented strategy opens a wide area of numerical applications for hardware acceleration ***(first page, Abstract, first paragraph),*** and turns a graphics card into an ultrafast vector coprocessor ***(first page, Abstract, first paragraph)*** which would have been recognized by the ordinary artisan as benefits that allow faster processing." Applicant respectfully disagrees with this proposition, especially in view of the vast evidence to the contrary.

e. For example, Press relates to implementing mathematics in software, while Rumpf relates to using graphics cards for quantized FEM computations. To simply glean features from a system for performing quantized FEM computations using graphics cards, such as that of Rumpf, and combine the same with the *non-analogous art* of software-implemented mathematics such as that of Press, would simply be improper. Graphics cards provide broad access to graphics memory and parallel processing of image operands (see the Abstract of Rumpf), while software-implemented mathematics merely relates to using software to carry out mathematical operations. "In order to rely on a reference as a basis for rejection of an applicant's invention, the reference must either be in the field of applicant's endeavor or, if not, then be reasonably pertinent to the particular problem with which the inventor was concerned." In re Oetiker, 977 F.2d 1443, 1446, 24 USPQ2d 1443, 1445 (Fed. Cir. 1992). See also In re Deminski, 796 F.2d 436, 230 USPQ 313 (Fed. Cir. 1986); In re Clay, 966 F.2d 656, 659, 23 USPQ2d 1058, 1060-61 (Fed. Cir. 1992). In view of the vastly different types of problems software-implemented mathematics addresses as opposed to graphics cards, the Examiner's proposed combination is clearly inappropriate.

i. The Examiner respectfully replies:

ii. The Applicant appears to start by disagreeing with the motivation to combine the references, and then appears to shift to asserting that Rumpf and Press are not

analogous art. As recited above, "the motivation to use the art of Rumpf with the art of Press would have been the benefits recited in Rumpf that the presented strategy opens a wide area of numerical applications for hardware acceleration ***(first page, Abstract, first paragraph),*** and turns a graphics card into an ultrafast vector coprocessor ***(first page, Abstract, first paragraph)*** which would have been recognized by the ordinary artisan as benefits that allow faster processing". Since this motivation is taken from the reference, and has obvious benefits of saving computation time, the motivation appears acceptable.

iii. Regarding the assertion that Rumpf and Press are not analogous art, the art of Rumpf and the art of Press are analogous art at least because they both pertain to solving partial differential equations, which is "reasonably pertinent to the particular problem with which the inventor was concerned". Further, the ordinary artisan would have known that software is at least implemented on hardware for use, and so is reasonably pertinent.

f. The Applicant argues:

g. In addition, applicant respectfully asserts that the software mathematics of the Press and Press2 references are implemented using "Fortran 77" and "C" (see respective Titles), but are not disclosed to be directly on a graphics card. For example, Page 860 of Press discloses implementing "a routine for SOR with Chebyshev acceleration" in Fortran.

i. The Examiner respectfully replies:

ii. While the assertion appears to be correct, no conclusions appear to be made from the assertion.

h. The Applicant argues:

i. Further, Rumpf discloses having to "approximate all involved nonlinear functions by linear in the implementation of the anisotropic diffusion" which "leads to an deterioration in image quality in the following timesteps," and that "the restricted precision of bits per color component leads to unsatisfying results for the linear heat equation... with very high relative errors" (Section 8, second column,

paragraph 3 —emphasis added). Again, applicant respectfully asserts that the Examiner's proposed combination is clearly inappropriate in view of the vastly different types of problems addressed by software-implemented mathematics as opposed to those addressed by graphics card-implemented quantized FEM computations.

i. The Examiner respectfully replies:

ii. As recited above, "Further, Rumpf discloses having to "approximate all involved nonlinear functions by linear in the implementation of the anisotropic diffusion" which "leads to an deterioration in image quality in the following timesteps," and that "the restricted precision of bits per color component leads to unsatisfying results for the linear heat equation... with very high relative errors" (Section 8, second column, paragraph 3 — emphasis added)", but this appears to be a premise, and does not appear to result in the conclusion that, "the Examiner's proposed combination is clearly inappropriate in view of the vastly different types of problems addressed by software-implemented mathematics as opposed to those addressed by graphics card-implemented quantized FEM computations".

iii. Further, the problems addressed by the software of Press appear to be similar to the problem addressed by graphics card implemented quantized FEM computations at least because they both pertain to solving partial differential equations, and further Rumpf teaches using a Jacobi solver relaxation method (Section 8), and Press also teaches using the Jacobi method in section 19.5 Relaxation Methods for Boundary Value Problems.

iv. Further, a new reference, "Diffusion models and their accelerated solution in image and surface processing" appears to teach that graphics cards execute commands from memory (*page 26, section 5, first paragraph*) used to solve partial differential equation on a graphics processor.

j. The Applicant argues:

k. Furthermore, Rumpf discloses that "many numerical algorithms still disregard hardware issues and little humps in the graphics hardware still **obstruct the Passage to general fast numerical computations"** (Section 1,

Goals, paragraph 4 - emphasis added). Applicant asserts that Rumpf's disclosure that many algorithms disregard hardware issues and that graphics hardware obstructs passage to general fast numerical computations clearly *teaches away* from the software-implemented general mathematics of the Press references. *In re Hedges. 783 F.2d 1038, 228 USPQ 685 (Fed. Cir. 1986).*

i. The Examiner respectfully replies:

ii. Rumpf also teaches in the same section, "Certainly there are some obstacles . . . but the overall hardware design and development amazingly fits the numerical purpose," which clearly indicates an expectation of successful implementation. Further, since Rumpf actually implemented the linear equation solver on the graphics card, any concerns over success were certainly overcome.

l. The Applicant argues:

m. In the Office Action mailed 07/12/2007, the Examiner has argued that "the art of Rumpf and the art of Press are clearly analogous art for at least the reason that they both pertain to the art of solving partial differential equations *(Press. page 838, section 19.2, Diffusive Initial Value Problems: and Rump! section 6. Linear Heat Equation)."* Further, the Examiner has argued that "Rumpf continues, 'Hence even minor considerations of graphics hardware issues with respect to numerics on one side, and development of slightly more hardware sensitive algorithms on the other, could **result in revolutionary speedups for many applications'"** such that "Rumpf is clearly advocating the method for improving performance of applications."

n. Applicant respectfully disagrees and asserts that Press clearly teaches software implemented mathematics for diffusive initial value problems (Page 838, section 19.2), whereas Rumpf teaches "a graphics hardware solver for the linear heat equation" (Section 6 - emphasis added). Furthermore, Rumpf discloses that "many numerical algorithms still disregard hardware issues and little humps in the graphics hardware still **obstruct the passage to general fast numerical computations,"** where "minor considerations of graphics hardware issues with respect to numerics, and the development of slightly more hardware

sensitive algorithms on the other, could result in revolutionary speedups for many applications" (Section 1, Goals, paragraph 4 - emphasis added). Thus, as expressly taught by Rumpf, many numerical algorithms still disregard hardware issues, and the **development of** slightly more hardware sensitive algorithms could result in revolutionary speedups for many applications, which clearly *leaches away* from using general mathematics which are not disclosed to even take into account hardware considerations, such as those originally disclosed by Press in 1986. Therefore, Rumpf advocates the development of more hardware sensitive algorithms, which fails to support the Examiner's proposed combination of combining the software implemented mathematics of Press with the graphics hardware implemented quantized FEM computations of Rumpf.

i. The Examiner respectfully replies:

ii. While Rumpf teaches a graphics hardware solver for the linear heat equation, the hardware is being used to implement an algorithm (section 6, pseudo code) using instructions (section 2 Computational Setting, first paragraph, "our applications are based on the OpenGL API [12], and to provide accuracy we will refer to the OpenGL commands in the text"). Regarding teaching away, since Rumpf actually implemented the linear equation solver on the graphics card, any concerns over success were certainly overcome.

o. The Applicant argues:

p. Also in the Office Action mailed 07/12/2007, the Examiner has argued that "hardware and software are equivalent (please refer to the new reference by Tanenbaum, Structured Computer Organization, 1984, page 11)." Applicant respectfully disagrees and asserts that the Examiner's reliance upon the Tanenbaum reference constitutes a reference(s) separate from those in the relevant rejection under 35 U.S.C. 103(a). Further, it is noted that the Examiner has failed to cite specific motivation in the relevant reference(s) to support the case for combining the Press, Rumpf, and Tanenbaum reference(s). The Examiner is reminded that the Federal Circuit requires that there must be some

logical reason apparent from the evidence of record that would justify the combination or modification of references. *In re Regel,* 188 USPQ 132 (CCPA 1975). Thus, the reliance on the Tanenbaum reference(s), on its face, is clearly improper.

i. The Examiner respectfully replies:

ii. In the context of the previous Office Action, it appears that the reference to Tanenbaum is not strictly needed to overcome the Applicant's argument, and so the reference is withdrawn except as teaching knowledge of the ordinary artisan at the time of invention that hardware and software were equivalent.

q. The Applicant argues:

r. To this end, at least the first element of the *prima facie* case of obviousness has not been met, since it would be *unobvious* to combine the references, as noted above.

i. The Examiner respectfully replies:

ii. As discussed above, the Examiner maintains that it would have been obvious to combine the references.

s. The Applicant argues:

t. More importantly, with respect to the third element of the *prima facie* case of obviousness, the Examiner has relied upon Page 395, right-side column, and Page 400, left-side column from Roy-Chowdhury, in addition to Page 855, second paragraph from Press, to make a prior art showing of applicant's claimed technique "wherein the determining whether the solution has converged includes calculating errors and concluding that the solution has converged based on the calculation of the errors" (see this or similar, but not necessarily identical language in independent Claims 1, 10, 11, 26, 27, 28, and 30). Furthermore, in the Office Action mailed 07/12/2007, the Examiner has stated, in Section vii on Pages 13 and 14, that Press does not specifically teach applicant's claimed technique.

u. Applicant respectfully asserts that the excerpt from Press relied upon by the Examiner merely discloses that "the algorithm consists of using the average of *u* at its four nearest-neighbor points on the grid" which "is then iterated until convergence" (Page 855). Further, the excerpt from Press discloses this method as the "classical method... called *Jacobi's method*" (Page 855). However, the mere disclosure of Jacobi's method which iterates the use of the average of *u* at its four nearest-neighbor points until convergence, as in Press, simply fails to even suggest a technique "wherein the determining whether the solution has converged includes calculating errors and concluding that the solution has converged based on the calculation of the errors" (emphasis added), as claimed by applicant. Clearly, Jacobi's method, as in Press, simply fails to suggest "calculating errors," let alone "concluding that the solution has converged based on the calculation of the errors" (emphasis added), as claimed by applicant.

i. The Examiner respectfully replies:

ii. With this Office Action, the Examiner is removing the reference to Roy-Chowdhury regarding the recited limitation, and is relying upon a new reference by Burden. Please refer to the updated rejection below.

v. The Applicant argues:

w. Further, applicant respectfully asserts that the excerpts from Roy-Chowdhury relied upon by the Examiner merely disclose that "[t]he expressions for updating errSR_ and errSB_ in each iteration... may be derived by summing over all red and black points" (Page 400). Further, the excerpts from Roy-Chowdhury disclose that "wherever error bounds for individual elements of u[i][j] arise in our error expressions, we drop them" (Page 400). However, the mere disclosure of updating errSR_ and errSB_ in each iteration, and dropping error bounds for individual elements when they arise in the error expressions, as in Roy-Chowdhury, simply fails to suggest a technique "wherein the determining whether the solution has converged includes calculating errors and concluding that the solution has converged based on the calculation of the errors" (emphasis added), as claimed by applicant.

x. Furthermore, applicant notes that Row-Chowdhury discloses that "[i]n this paper, we develop low-overhead, error-detecting versions of iterative algorithms for solving the regular, spare linear systems which arise from discretizations of various partial differential equations (PDEs)" (Page 394, second column - emphasis added). Clearly, disclosing error-detecting versions of iterative algorithms, as in Row-Chowdhury, simply fails to even suggest that "determining whether the solution has converged includes calculating errors and concluding that the solution has converged based on the calculation of the errors" (emphasis added), in the manner as claimed by applicant.

y. In the Office Action mailed 07/12/2007, the Examiner has argued that "[Ole excerpt from Roy-Chowdhury needs to be evaluated in the context of the reference, especially the teaching that the termination condition for the iterative method is determined at runtime by specifying that the outer loop continue until the maximum difference over all grid points of a point value at the current iteration from its value at a previous iteration drops below a threshold *(page 395, right-side column, top halt)*."

z. Applicant respectfully disagrees and asserts that Page 395, left column, last paragraph, from Roy-Chowdhury merely discloses that "[w]e may 'solve' the Laplace equation numerically over a region by discretizing it in the x and y directions to obtain a grid of points and then computing the approximate solution values at these points," such that "[t]he termination condition is determined at runtime by specifying that the outer loop continue until the maximum difference over all grid points of a point value at the current iteration from its value at the previous iteration drops below a threshold" (emphasis added). However, the mere disclosure of solving the Laplace equation by obtaining a grid of points and approximating a solution at these points until a point value in the current iteration drops below a threshold from its value in the previous iteration as in Roy-Chowdhury, simply fails to even suggest that "determining whether the solution has converged includes calculating errors and concluding that the solution has converged based on the calculation of the errors" (emphasis added), in the manner as claimed by applicant. Clearly, simply teaching that a point value in the current iteration drops below a threshold from its value in the previous

iteration, as in Roy-Chowdhury, fails to suggest "calculating errors," let alone "concluding that the solution has converged based on the calculation of the errors" (emphasis added), as claimed by applicant.

aa. Further, in the Office Action mailed 07/12/2007, the Examiner has argued that "[i]n combination with the context, the disclosure for updating errSR_ and errSB_ in each iteration suggests [that] determining whether the solution has converged includes concluding that the solution has converged based on the calculation of said errors."

bb. Applicant strongly disagrees, and asserts that Section 2.4 starting on Page 399 of Roy-Chowdhury describes the "Modified Algorithm with Checks and Error Bounding." Further, the sample code on the right column of Page 399 clearly illustrates that, inside the for loop "for (k^r,k, k<iter; k++)" (emphasis added), the red points are updated, then the red sums and error variables (SR, errSk, errSR) are updated, and then the black points and black sums and error variables (SB, errSB_, errSB) are updated. Further, after the "for loop" iterates for "iter" iterations and completes (emphasis added), the error detection code then "check[s the] sum of the red points" and "check[s the] sum of the black points." Applicant notes that the error detecting code occurs after the "for loop" has iterated for "iter" iterations.

cc. However, the error detecting SOR code that first iterates for iter iterations updating errSR_ and errSB_, and then, after the for loop completes, checks for errors and performs necessary "error()" handling functions, as in Roy-Chowdhury, simply fails to even suggest that "determining whether the solution has converged includes calculating errors and concluding that the solution has converged based on the calculation of the errors" (emphasis added), in the manner as claimed by applicant. Clearly, iterating for iter iterations and then checking for error conditions, as in Roy-Chowdhury, simply fails to suggest **"determining** whether the solution has converged includes calculating errors" and "concluding that the solution has converged based on the calculation of the errors" (emphasis added), in the manner as claimed by applicant.

i. The Examiner respectfully replies:

ii. Applicant's arguments are persuasive, and new rejections are made below.

dd. The Applicant argues:

ee. Still yet, in the Office Action mailed 07/12/2007, the Examiner has argued that "the specification appears to be silent on the meaning of 'calculating errors."' Applicant respectfully disagrees, and asserts that applicant's specification, as originally filed, is not silent on "calculating errors," as alleged by the Examiner. For example, see Page 5, lines 4-9; Page 13, lines 4-8; and Page 17, lines 12-29 et al. Of course, such citations (in combination with the remaining specification) are merely examples of the above claim language and should not be construed as limiting in any manner.

i. The Examiner respectfully replies:

ii. While the specification is not silent on "calculating errors", the specification appears to be silent on "the meaning of calculating errors". In the specification, page 17, line 12, errors appear to mean "residuals", but the meaning of residuals is not defined (residuals of what?), nor does an example appear to be provided.

ff. The Applicant argues:

gg. Still yet, with respect to independent Claim 29, applicant respectfully asserts that such claim is deemed novel in view of the prior art excerpts relied on by the Examiner for at least substantially the same reasons argued above. For example, Claim 29 recites "determining whether the solution has converged by: calculating errors, summing the errors, and concluding that the solution has converged if the sum of errors is less than a predetermined amount" (emphasis added), as claimed, which is clearly not met by the prior art excerpts relied on by the Examiner, for substantially the same reasons as noted above.

hh. To this end, applicant respectfully asserts that at least the first and third elements of the *prima facie case* of obviousness have not been met, since it would be *unobvious* to combine the references, as noted above, and the prior art excerpts, as relied upon by the Examiner, fail to teach or suggest all of the claim limitations, as noted above. Thus, a notice of allowance or a proper prior art showing of all of

applicant's claim limitations, in combination with the remaining claim elements, is respectfully requested.

ii.

i. The Examiner respectfully replies:

ii. Applicant's argument regarding claim 29 is persuasive, however a new rejection is made below.

jj. The Applicant argues:

kk. Applicant further notes that the prior art is also deficient with respect to the dependent claims. For example, with respect to Claim 13, the Examiner has relied upon Pages 854-856 in Press to make a prior art showing of applicant's claimed technique "wherein the relaxation operation is selected based on the partial differential equation." Specifically, the Examiner has argued that "it would have been obvious that the relaxation operation is selected on the partial differential equation, especially since such an example is presented."

ll. Applicant respectfully disagrees and asserts that the excerpt from Press relied upon by the Examiner merely discloses that "relaxation methods involve splitting the sparse matrix that arises from finite differencing and the iteration until a solution is found" (Page 854). For example, Press discloses a "method... called Jacobi's method" which "is not practical because it converges too slowly," in addition to "[t]he Gauss-Seidel method" which offers a "factor of two improvement in the number of iterations over the Jacobi method [which] still leaves the method impractical" (Pages 854-857). Clearly, Press is merely disclosing two different classical relaxation methods, the Jacobi's method and the Gauss-Seidel method, which clearly fails to support the Examiner's allegation that "it would have been obvious that the relaxation operation is selected on the partial differential equation," especially in view of applicant's claimed technique, namely "wherein the <u>relaxation operation</u> is <u>selected</u> based on the <u>partial differential equation"</u> (emphasis added), as claimed by applicant.

mm. To this end, in response to the Examiner's argument that applicant's specific claim language would have been obvious, applicant

again points out the remarks above that clearly show the manner in which some of such claims further distinguish Press. Applicant thus formally requests a specific showing of the subject matter in ALL of the claims in any future action. Note excerpt from MPEP below.

nn. "If the applicant traverses such an [Official Notice] assertion the examiner should cite a reference in support of his or her position." See MPEP 2144.03.

i. The Examiner respectfully replies:

ii. While the Examiner appreciates the Applicant's argument, the recited art appears to teach the limitation, as follows. Pages 854 – 856 appear to teach the limitation that the relaxation operation is selected based on the partial differential equation because, on page 855, the diffusion equation is recited in equation 19.5.3, and then a relaxation method of equation 19.5.5 is derived from the diffusion equation, and as recited below equation 19.5.5, "Thus the algorithm consists of using the average of u at its four nearest-neighbor points on the grid (plus the contribution from the source). This procedure is then iterated until convergence". Thus, the relaxation method appears to have been selected based on the partial differential equation (the diffusion equation).

iii. Further, knowledge of the ordinary artisan to select a numerical method for a partial differential equation is taught in the reference, J.L. Bell and G.S. Patterson Jr., "Data organization in large numerical computations", The Journal of Supercomputing, Volume 1, Number 1, page 130, last paragraph, and page 132, figure 13.

oo. The Applicant argues:

pp. Further, with respect to Claim 18, the Examiner has relied on Page 855 from the Press reference to make a prior art showing of applicant's claimed technique "wherein it is determined whether the solution has converged after a predetermined number of multiple iterations of the relaxation operation." Further, the Examiner has stated that Press does not specifically teach that "[i]t is determined whether the solution has converged ***after a predetermined number of multiple*** iterations of the relaxation operation." Additionally, the Examiner has relied upon Official Notice and has stated that "processing time would be saved by testing convergence only after multiple iterations for a process that

takes multiple iterations to converge." Specifically, as support for Official Notice, the Examiner has relied upon Galligani et al. ("Implementation of Splitting Methods for Solving Block Tridiagonal Linear Systems on Transputers"), Beckmann et al. ("Data Distribution at Run-Time: Re-Using Execution Plans"), and Y. Saad ("Krylov Subspace Methods for Solving Large Unsymmetric Linear Systems").

qq. Applicant respectfully disagrees and asserts that the Examiner's reliance upon the Galligani, Beckmann, and Saad references constitutes a reference(s) separate from those in the relevant rejection under 35 U.S.C. 103(a). Further, it is noted that the Examiner has failed to cite specific motivation in the relevant reference(s) to support the case for combining the Galligani, Beckmann, and Saad reference(s). The Examiner is reminded that the Federal Circuit requires that there must be some logical reason apparent from the evidence of record that would justify the combination or modification of references. In re Regel, 188 USPQ 132 (CCPA 1975). Thus, the reliance on the Galligani, Beckmann, and Saad reference(s), on its face, is clearly improper.

rr. In view of the Examiner's improper reliance on the Galligani, Beckmann, and Saad reference(s), and in response to the Examiner's reliance on Official Notice, applicant respectfully asserts that, in view of Press, it would not have been obvious to "[determine] whether the solution has converged after a predetermined number of multiple iterations of the relaxation operation," as claimed. Specifically, the excerpt from Press disclosing "interat[ion] until convergence," as relied on by the Examiner, merely relates to "a classical method with origins dating back to the last century, called *Jacobi's method*" (Page 855), which does not even suggest "a predetermined number of multiple iterations," as claimed. Thus, applicant again formally requests a specific showing of the subject matter in ALL of the claims in any future action (MPEP 2144.03).

i. **The Examiner respectfully replies:**

ii. **The Examiner respectfully asserts that using the references of Galligani, Beckmann, and Saad to support the Official Notice is proper. The use of the references appears to comply with the MPEP 2144.03.**

iii. Further, the rejection specifically recites that the motivation to use the Official Notice would have been the knowledge of the ordinary artisan that processing time would be saved by testing convergence only after multiple iterations for a process that takes multiple iterations to converge.

iv. Further, the rejection has been amended such that Press is not relied upon to teach "a predetermined number of multiple iterations".

ss. The Applicant argues:

tt. In addition, with respect to Claim 21, the Examiner has relied on Page 395, right-side column, and Page 400, left-side column, from Roy-Chowdhury to make a prior art showing of applicant's claimed technique "wherein the determining whether the solution has converged further includes concluding that the solution has converged if an error is less than a predetermined amount."

uu. Applicant respectfully disagrees and asserts that Page 395 from Roy-Chowdhury merely discloses that "[vi]e may 'solve' the Laplace equation numerically over a region by discretizing it in the x and y directions to obtain a grid of points and then computing the approximate solution values at these points," such that "[t]he termination condition is determined at runtime by specifying that the outer loop continue until the maximum difference over all grid points of <u>a Point value at the current iteration from its value at the previous iteration drops below a threshold"</u> (emphasis added). Further, applicant respectfully asserts that the excerpts from Roy-Chowdhury relied upon by the Examiner merely disclose that "[t]he expressions for updating errSR and errSB_ in each iteration... may be derived by summing over all red and black points," and that "wherever error bounds for individual elements of u[i][j] arise in our error expressions, we drop them" (Page 400). Additionally, Page 395 in Roy-Chowdhury discloses that "[w]e <u>omit the convergence check</u> in subsequent discussions since it has <u>no bearing</u> on the development of the <u>error-detecting algorithm"</u> (emphasis added).

vv. However, the mere disclosure of solving the Laplace equation by obtaining a grid of points and approximating a solution at these points until a point value in the current iteration drops below a threshold from its value in the previous iteration, in addition to updating errSR

and errSB in each iteration, and dropping error bounds for individual elements when they arise in the error expressions, as in Roy-Chowdhury, simply fails to suggest a technique "wherein the determining whether the solution has converged further includes concluding that the solution has converged if an error is less than a predetermined amount" (emphasis added), as claimed by applicant. Clearly, iterating until a point value in the current iteration drops below a threshold of its value in the previous iteration, in addition to dropping error bounds when they arise, as in RoyChowdhury, fails to disclose "concluding that the solution has converged if an error is less than a predetermined amount" (emphasis added), as claimed by applicant. Further, Roy-Chowdhury's teaching that the convergence check is omitted since it has no bearing on the development of the error-detecting algorithm does not teach "concluding that the solution has converged if an error is less than a predetermined amount" (emphasis added), as claimed by applicant.

ww. Furthermore, applicant asserts that Section 2.4 starting on Page 399 of RoyChowdhury describes the "Modified Algorithm with Checks and Error Bounding." Further, the sample code on the right column of Page 399 clearly teaches that, inside the for loop "for (1.); k<iter k++)" (emphasis added), the red points are updated, then the red sums and error variables (SR, errSk, errSR) are updated, and then the black points and black sums and error variables (SB, errSB_, errSB) are updatec. Further, after the "for loop" iterates for `titer" iterations and completes (emphasis added), the error detection code then "check[s the] sum of the red points" and "check[s the] sum of the black points." Applicant notes that the error detecting code occurs after the "for loop" has iterated for "iter" iterations. Clearly, the Error detecting SOR code that first iterates for **iter** iterations updating errSR and errSB_, and then, after the for loop completes, checks for errors and performs necessary "error()" handling functions, as in RoyChowdhury, simply fails to even suggest a technique "wherein the determining whether the solution has converged further includes concluding that the solution has converged it' an error is less than a predetermined amount" (emphasis added), as claimed by applicant.

i. The Examiner respectfully replies:

ii. Applicant's argument is persuasive, however, a new rejection is made below.

xx. The Applicant argues:

yy. Still yet, with respect to Claim 22, the Examiner has relied on Pages 838-840 in Press to make a prior art showing of applicant's claimed technique "wherein if it is determined that the solution has converged, repeating the processing using an altered parameter value." Specifically, the Examiner has argued that "especially note on page 840 below equation 19.2.12, the reference to stepsize At," and that "[t]he specification appears to provide a time value as an example of a parameter on page 5, line 9."

zz. Applicant respectfully disagrees and asserts that the excerpt from Press teaches that in order "[t]o solve [the diffusion equation in one space dimension with a constant diffusion coefficient] one has to solve a set of simultaneous linear equations at each timestep for the u_j^{n+1}" (Pages 838 - 839). Further, the excerpt from Press teaches that "[t]he amplification factor for [the] equation (19.2.8) is... < 1 for any stepsize At" which "is unconditionally stable" (Page 840). However, the mere disclosure of solving a set of simultaneous linear equations at each timestep in order to solve the diffusion equation, as in Press, simply fails to even suggest a technique "wherein if it is determined that the **solution has converged,** repeating the processing using an altered parameter value" (emphasis added), as claimed by applicant. Clearly, solving a set of simultaneous linear equations at each timestep, as in Press, simply fails to even suggest that "if it is determined that **the solution has converged....the** processing [is repeated] using an altered parameter value," where the solution is "the solution to the partial differential equation" (emphasis added), in the context as claimed by applicant (see Claim 1 for context).

aaa. Again, since at least the first and third elements of the *prima facie case* of obviousness have not been met, as noted above, a notice of allowance or specific prior art showing of each of the foregoing claim elements, in combination with the remaining claimed features, is respectfully requested.

bbb. To this end, all of the independent claims are deemed allowable. Moreover, the remaining dependent claims are further deemed allowable, in view of their dependence on such independent claims.

i. The Examiner respectfully replies:

ii. The Examiner appreciates the argument, but respectfully asserts that the recited art appears to teach the limitation, as follows. The specification appears to provide a time value as an example of a parameter on page 5, line 9. In the recited art in Press at pages 838 – 840, time is clearly used to advance the solution after a set of linear equations are solved at a current timestep. The solution of the linear equations require convergence of the solution, and then time is advanced, And thus the altered parameter of time is used in the next timestep where a different set of linear equations are solved.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Regarding all **claims 1 - 2, 4 - 5, 7 - 15, 17 - 18 and 20 – 31**, the art of Rumpf (Martin Rumpf et al.; "Using Graphics Cards for Quantized FEM Computations") teaches using a graphics hardware pipeline to solve partial differential equations. After the inventive step of Rumpf, implementing any known method of solving a partial differential equation using a hardware graphics pipeline would have been obvious. The ordinary artisan would have known to turn to references describing solution methods to partial differential equations both by the nature of the problem and the benefit of saving time and cost by using proven previous solution methods. Further, the reference by Wang ("A Processor Architecture for 3D Graphics", September 1992, IEEE Computer Graphics & Applications) discloses a graphics pipeline with a suggestion that it can be used to solve partial differential equations (*page 97, first column*). Further, since a graphics pipeline performs numeric calculation, it is inherent in the device that it can be used to solve a partial differential equation (see MPEP section 2112).

6. **Claims 1 - 2, 12 - 15, 17 - 18, 20 - 23 and 27** are rejected under 35 U.S.C. 103(a) as being unpatentable over Press (Press, William H.; Flannery, Brian P.; Teukolsky, Saul A.; Vetterling, William T.; "Numerical Recipes in Fortran 77", 2001, Second edition, Cambridge University Press) in view of Rumpf (Martin Rumpf et al.; "Using Graphics Cards for Quantized FEM Computations", September 3 - 5 2001, Proceedings of the IASTED International Conference on Visualization, Imaging and Image Processing), further in view of Burden (Richard L. Burden and J. Douglas Faires, "Numerical Analysis", fourth edition, 1989, PWS-Kent Publishing Company, pages 383 - 393, 400 - 403 and 605 - 643).

a. The art of Press is directed to numerical methods for solving partial differential equations (*page 838, section 19.2*).

b. The art of Rumpf is directed to the art of solving partial differential equations using a graphics card (*Title and Abstract*).

c. The art of Burden is directed to numerical analysis, including numerical solutions to partial differential equations (*title and pages 605 - 643*).

d. The art of Rumpf and the art of Press are analogous art at least because they are both pertain to the art of solving partial differential equations.

e. The art of Rumpf and the art of Burden are analogous art at least because they are both pertain to the art of solving partial differential equations.

f. Regarding **claim 1**:

g. Press appears to teach:

i. Receiving input (**pages 854-856, section 19.5 Relaxation Methods for Boundary Value Problems; it would have been obvious that input is required to solve a partial differential equation, especially given the statement that an initial distribution relaxes to an equilibrium distribution on page 855**);

ii. Processing the input to generate the solution to the partial differential equation (**pages 854-856, section 19.5 Relaxation Methods for Boundary Value Problems**);

iii. Wherein the processing further includes determining whether the solution has converged (**pages 855, Relaxation Methods for Boundary Value Problems; second paragraph, section that starts with "Thus the algorithm consists . . .", sentence, "This procedure is then iterated until convergence."**);

h. Press does not specifically teach:

i. Receiving input ***in the hardware graphics pipeline***;

ii. Processing the input to generate the solution to the partial differential equation ***utilizing the hardware graphics pipeline***;

iii. ***Generating output utilizing the hardware graphics pipeline for display***;

iv. ***Wherein the solution to the partial differential equation is generated utilizing the hardware graphics pipeline for enhancing graphics processing operations performed by the hardware graphics pipeline***.

v. ***Wherein the graphics processing operations performed by the hardware graphics pipeline are enhanced by determining a location of surfaces or objects for rendering purposes utilizing the solution to the partial differential equation generated utilizing the hardware graphics pipeline***.

vi. ***Wherein the input includes a local area of textures used to sample a texture map to generate a modified local area of textures***;

vii. ***Wherein the determining whether the solution has converged includes calculating errors and concluding that the solution has converged based on the calculation of the errors***.

i. Rumpf appears to teach:

i. Receiving input in the hardware graphics pipeline (***third page, figure 1***);

ii. Processing the input to generate the solution to the partial differential equation utilizing the hardware graphics pipeline (***third page, section 3.1 Vector Representation, first paragraph; and seventh page, section 6. Linear Heat Equation, first paragraph***);

iii. Generating output utilizing the hardware graphics pipeline for display (***ninth page, figure 3***);

iv. Wherein the solution to the partial differential equation is generated utilizing the hardware graphics pipeline for enhancing graphics processing operations performed by the hardware graphics pipeline (***seventh page, section 6. Linear Heat Equation, first paragraph***);

v. Wherein the graphics processing operations performed by the hardware graphics pipeline are enhanced by determining a location of surfaces or objects for rendering purposes utilizing the solution to the partial differential equation generated utilizing the

hardware graphics pipeline (***ninth page, figure 3, displays surfaces and objects rendered by utilizing the solution to a partial differential equation utilizing a hardware graphics pipeline***).

vi. Wherein the input includes a local area of textures used to sample a texture map to generate a modified local area of textures (***second and third pages, section 2. Computational Setting; and third page, figure 1***).

j. Burden appears to teach:

i. Wherein the determining whether the solution has converged includes calculating errors and concluding that the solution has converged based on the calculation of the errors (***page 403, Jacobi Iterative Algorithm 7.1, step 4,if | |x - XO| | < TOL then OUTPUT(x1, . . . xn); STOP.; where | |x-XO| | calculates errors x-XO and takes the norm, see pages 384 - 392, and especially page 393, exercise 2 which defines a norm that sums the vector components***).

k. The motivation to use the art of Rumpf with the art of Press would have been the benefits recited in Rumpf that the presented strategy opens a wide area of numerical applications for hardware acceleration (***first page, Abstract, first paragraph***), and turns a graphics card into an ultrafast vector coprocessor (***first page, Abstract, first paragraph***), which would have been recognized by the ordinary artisan as benefits that allow faster processing.

l. The motivation to use the art of Burden with the art of Press would have been the benefit recited in Burden that iterative techniques are efficient in terms of computational time and computer storage for large systems that have a high percentage of zero entries which arises frequently in numerical solution of partial differential equations (***pages 400 - 401, last paragraph of page 400 continued on page 401***).

m. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Rumpf and the art of Burden with the art of Press to produce the claimed invention.

===

n. Regarding **claim 2**:

o. Press appears to teach:

i. Input represents boundary conditions (**pages 854-856, section 19.5 Relaxation Methods for Boundary Value Problems; it would have been obvious that boundary conditions are required to solve a partial differential equation, especially since the title of the section recites Boundary Value problems**);

==

p. Regarding **claim 12**:

q. Press appears to teach:

i. The processing includes a relaxation operation (**pages 854-856, section 19.5 Relaxation Methods for Boundary Value Problems; it would have been obvious that processing includes a relaxation operation, especially since the title of the section recites Relaxation Methods**);

==

r. Regarding **claim 13**:

s. Press appears to teach:

i. The relaxation operation is selected based on the partial differential equation (**pages 854-856, section 19.5 Relaxation Methods for Boundary Value Problems; it would have been obvious that the relaxation operation is selected based on the partial differential equation, especially since such an example is presented; on page 855, the diffusion equation is recited in equation 19.5.3, and then a relaxation method of equation 19.5.5 is derived from the diffusion equation, and as recited below equation 19.5.5, "Thus the algorithm consists of using the average of u at its four nearest-neighbor points on the grid (plus the contribution from the source). This procedure is then iterated until convergence". Thus, the relaxation method appears to have been selected based on the partial differential equation (the diffusion equation)**);

==

t. Regarding **claim 14**:

u. Press appears to teach:

i. The processing includes a plurality of iterations of the relaxation operation (**<u>pages 854-856, section 19.5 Relaxation Methods for Boundary Value Problems; especially references to Gauss-Seidel method and Jacobi's method</u>**);

==

v. Regarding **claim 15**:

w. Press appears to teach:

i. A number of iterations of the relaxation operation is reduced using at least one of a prolongation operation and a restriction operation (**<u>pages 862-868, section 19.6 Multigrid Methods for Boundary Value Problems, especially page 865 Smoothing, Restriction and Prolongation Operators</u>**);

==

x. Regarding **claim 17**:

y. Press appears to teach:

i. It is determined whether the solution has converged after each iteration of the relaxation operation (**<u>pages 855, Relaxation Methods for Boundary Value Problems; second paragraph, section that starts with "Thus the algorithm consists . . .", sentence, "This procedure is then iterated until convergence."</u>**);

==

z. Regarding **claim 18**:

aa. Press appears to teach:

i. It is determined whether the solution has converged after iterations of the relaxation operation (**<u>page 855, Relaxation Methods for Boundary Value Problems; second paragraph, section that starts with "Thus the algorithm consists . . .", sentence, "This procedure is then iterated until convergence."</u>**);

bb. Press does not specifically teach:

i. It is determined whether the solution has converged ***after a predetermined number of multiple*** iterations of the relaxation operation.

cc. Official Notice is taken that determining whether a solution has converged after a predetermined number of multiple iterations was old and well known at the time of invention in the analogous art of numerical methods. At the time of invention it would have been obvious to a person of ordinary skill in the art to determine whether the solution has converged ***after a predetermined number of multiple*** iterations of the relaxation operation. The motivation would have been the knowledge of the ordinary artisan that processing time would be saved by testing convergence only after multiple iterations for a process that takes multiple iterations to converge. As support for the Official Notice, three references are provided:

i. E. Galligani et al.; "Implementation of Splitting Methods for Solving Block Tridiagonal Linear Systems on Transputers", 1995, Proceedings of Euromicro Workshop on Parallel and Distributed Processing, pages 409 – 415, especially page 412, left-side column, sentence that starts with, "The overheads can be minimized . . ."

ii. Olav Beckmann et al.; "Data Distribution at Run-Time: Re-Using Execution Plans", 1998, Euro-Par'98, LNCS 1470, Springer-Verlag, pages 413 – 421, especially page 418, text for Table 1, convergence test every 10 iterations.

iii. Y. Saad; "Krylov Subspace Methods for Solving Large Unsymmetric Linear Systems", July 1981, Mathematics of Computation, Volume 37, Number 155, pages 105 - 126; teaches testing for convergence every q steps, page 113.

dd. Therefore, it would have been obvious to modify Press and Rumpf to obtain the invention as specified in claim 18.

==

ee. Regarding **claim 20**:

ff. Press does not specifically teach:

i. The determining whether the solution has converged further includes summing the errors;

gg. Burden appears to teach:

i. The determining whether the solution has converged further includes summing the errors (*page 403, Jacobi Iterative Algorithm 7.1, step 4,if ||x - XO|| < TOL then OUTPUT(x1, . . . xn); STOP.; where ||x-XO|| calculates errors x-XO and takes the norm, see pages 384 - 392, and especially page 393, exercise 2 which defines a norm that sums the vector components*);

==

hh. Regarding **claim 21**:

ii. Press does not specifically teach:

i. Concluding that the solution has converged if an error is less than a predetermined amount;

jj. Burden appears to teach:

i. Concluding that the solution has converged if an error is less than a predetermined amount (*page 403, Jacobi Iterative Algorithm 7.1, step 4,if ||x - XO|| < TOL then OUTPUT(x1, . . . xn); STOP.; where TOL is a predetermined amount and where ||x-XO|| calculates errors x-XO and takes the norm, see pages 384 - 392, and especially page 393, exercise 2 which defines a norm that sums the vector components*);

==

kk. Regarding **claim 22**:

ll. Press appears to teach:

i. If it is determined that the solution has converged repeating the processing using an altered parameter value operation (**pages 838-840, section 19.2 Diffusive Initial Value Problems; especially note on page 840 below equation 19.2.12, the reference to stepsize Δt. The specification appears to provide a time value as an example of a parameter on page 5, line 9**);

==

mm. Regarding **claim 23**:

nn. Press appears to teach:

i. The number of iterations of the relaxation operation is determined prior to the processing (**<u>pages 860, Relaxation Methods for Boundary Value Problems; code example with a parameter value MAXITS = 1000 and a loop DO N=1,MAXITS</u>**);

==

oo. Regarding **claim 27**:

i. Claim 27 is taught as in claim 1 above.

==

7. **Claims 4 - 5** are rejected under 35 U.S.C. 103(a) as being unpatentable over Press as modified by Rumpf and Burden as applied to claims **1 - 2, 12 - 15, 17 - 18, 20 - 23 and 27** above, further in view of Weiskopf (Weiskopf, Daniel; Hopf, Matthias; Ertl, Thomas; "Hardware-Accelerated Visualization of Time-Varying 2D and 3D Vector Fields by Texture Advection via Programmable Per-Pixel Operations", 2001, Proceedings of the Vision Modeling and Visualization Conference 2001).

a. Press as modified by Rumpf and Burden teaches a hardware graphics pipeline implemented method for generating a solution to a partial differential equation in a hardware graphics pipeline.

b. The art of Weiskopf is directed to hardware accelerated visualization of time-varying 2D and 3D vector fields by texture advection (*<u>Title</u>*).

c. The art of Weiskopf and the art of Press as modified by Rumpf and Burden are analogous art because they both contain the art of performing calculations using a graphics card (*<u>Rumpf, Abstract; Weiskopf, pages 668 - 670, section 3.1Basic Advection</u>*).

d. The motivation to use the art of Weiskopf with the art of Press as modified by Rumpf and Burden would have been the benefits recited in Weiskopf including the advantage that the

algorithm has extremely high simulation and rendering speed (*page 672, right-side column, fourth paragraph that starts with, "An advantage . . ."*).

e. Regarding **claim 4**:

f. Press does not specifically teach:

i. the input includes geometry;

g. Weiskopf appears to teach:

i. the input includes geometry (**pages 668 - 669, section 3 Hardware-Based 2D Texture Advection; it would have been obvious that the input includes geometry; please note that the partial differential equation on page 667, right-side column, second paragraph, is being solved**);

h. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Weiskopf with the art of Press as modified by Rumpf and Burden to produce the claimed invention.

==

i. Regarding **claim 5**:

j. Press does not specifically teach:

i. the geometry is selected from the group consisting of polygons, vertex data, points, and lines;

k. Weiskopf appears to teach:

i. the geometry includes points (**pages 668 - 669, section 3 Hardware-Based 2D Texture Advection; it would have been obvious that the input includes geometry; please note that the partial differential equation on page 667, right-side column, second paragraph, is being solved**);

==

8. **Claims 7 – 9 and 24 - 25** are rejected under 35 U.S.C. 103(a) as being unpatentable over Press as modified by Rumpf and Burden as applied to claims **1 – 2, 12 – 15, 17 - 18, 20 – 23 and 27** above, further in view of Ewins (Ewins, Jon P.; Waller, Marcus D.; White, Martin; Lister, Paul F.; "MIP-Map Level Selection for Texture Mapping", 1998, IEEE Transactions on Visualization and Computer Graphics, Vol. 4, No. 4).

a. Press as modified by Rumpf and Burden teaches a hardware graphics pipeline implemented method for generating a solution to a partial differential equation in a hardware graphics pipeline.

b. The art of Ewins is directed to MIP-Map selection for texture mapping (*Title*).

c. The art of Ewins and the art of Press as modified by Rumpf and Burden are analogous art because they both pertain to the art of computer graphics generation (*Ewins, Abstract; Rumpf, figure 3 and section 7, Anisotropic Diffusion in Image Processing*).

d. Regarding **claim 7**:

e. Press does not specifically teach:

i. the local area of textures is generated by sampling a texture map;

f. Ewins appears to teach:

i. sampling a texture map (**pages 318 - 319, section 1.1 Texture Filtering**);

g. The motivation to use the art of Ewins with the art of Press as modified by Rumpf and Burden would have been the benefit recited in Ewins that texture mapping allows a high degree of visual complexity without the expense of overly complex geometric modeling (**page 317, section 1 Introduction, and Abstract**), which would have been recognized as a benefit by the ordinary artisan.

h. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Ewins with the art of Press as modified by Rumpf and Burden to produce the claimed invention.

==

i. Regarding **claim 8**:

j. Press does not specifically teach:

i. the local area of textures is filtered;

k. Ewins appears to teach:

i. the local area of textures is filtered (**pages 318 - 319, section 1.1 Texture Filtering**);

==

l. Regarding **claim 9**:

m. Press does not specifically teach:

i. the local area of textures is filtered utilizing a plurality of filters;

n. Ewins appears to teach:

i. the local area of textures is filtered utilizing a plurality of filters (**<u>pages 318 - 319, section 1.1 Texture Filtering</u>**);

==

o. Regarding **claim 24**:

p. Press does not specifically teach:

i. the filtering is carried out using a programmable filter;

q. Ewins appears to teach:

i. the filtering is carried out using a programmable filter (**<u>pages 318 - 319, section 1.1 Texture Filtering</u>**);

==

r. Regarding **claim 25**:

s. Press does not specifically teach:

i. the filtering is carried out using a non-programmable filter;

t. Ewins appears to teach:

i. the filtering is carried out using a non-programmable filter (**<u>pages 318 - 319, section 1.1 Texture Filtering</u>**);

==

9. **Claims 10 - 11** are rejected under 35 U.S.C. 103(a) as being unpatentable over Press (Press, William H.; Flannery, Brian P.; Teukolsky, Saul A.; Vetterling, William T.; "Numerical Recipes in Fortran 77", 2001, Second edition, Cambridge University Press) in view of Rumpf (Martin Rumpf et al.; "Using Graphics Cards for Quantized FEM Computations", September 3 – 5 2001, Proceedings of the IASTED International Conference on Visualization, Imaging and Image Processing), further in view of Burden

(Richard L. Burden and J. Douglas Faires, "Numerical Analysis", fourth edition, 1989, PWS-Kent Publishing Company, pages 383 – 393, 400 - 403 and 605 - 643).

a. Regarding **claim 10**:

b. Press appears to teach:

i. Receiving input (**pages 854-856, section 19.5 Relaxation Methods for Boundary Value Problems; it would have been obvious that input is required to solve a partial differential equation, especially given the statement that an initial distribution relaxes to an equilibrium distribution on page 855**);

ii. Processing the input to generate the solution to the partial differential equation (**pages 854-856, section 19.5 Relaxation Methods for Boundary Value Problems**);

iii. Wherein the processing further includes determining whether the solution has converged (**pages 855, Relaxation Methods for Boundary Value Problems; second paragraph, section that starts with "Thus the algorithm consists . . .", sentence, "This procedure is then iterated until convergence."**);

c. Press does not specifically teach:

i. Receiving input *in the hardware graphics pipeline*;

ii. Processing the input to generate the solution to the partial differential equation *utilizing the hardware graphics pipeline*;

iii. Generating output utilizing the hardware graphics pipeline for display;

iv. *Wherein the solution to the partial differential equation is generated utilizing the hardware graphics pipeline for enhancing graphics processing operations performed by the hardware graphics pipeline*;

v. *Wherein the graphics processing operations performed by the hardware graphics pipeline are enhanced by determining a location of surfaces or objects for rendering purposes utilizing the solution to the partial differential equation generated utilizing the hardware graphics pipeline*;

vi. *Wherein the input includes a local area of textures*;

vii. *Wherein the local area of textures is filtered utilizing a filter including a plurality of elements*;

viii. *Wherein the input includes a local area of textures used to sample a texture map to generate a modified local area of textures*;

ix. ***Wherein the determining whether the solution has converged includes calculating errors and concluding that the solution has converged based on the calculation of the errors***.

d. Rumpf appears to teach:

i. Receiving input in the hardware graphics pipeline (***third page, figure 1***);

ii. Processing the input to generate the solution to the partial differential equation utilizing the hardware graphics pipeline (***third page, section 3.1 Vector Representation, first paragraph; and seventh page, section 6. Linear Heat Equation, first paragraph***);

iii. Generating output utilizing the hardware graphics pipeline for display (***ninth page, figure 3***);

iv. Wherein the solution to the partial differential equation is generated utilizing the hardware graphics pipeline for enhancing graphics processing operations performed by the hardware graphics pipeline (***seventh page, section 6. Linear Heat Equation, first paragraph***);

v. Wherein the graphics processing operations performed by the hardware graphics pipeline are enhanced by determining a location of surfaces or objects for rendering purposes utilizing the solution to the partial differential equation generated utilizing the hardware graphics pipeline (***ninth page, figure 3, displays surfaces and objects rendered by utilizing the solution to a partial differential equation utilizing a hardware graphics pipeline***).

vi. Wherein the input includes a local area of textures (***third page, figure 1; please note the textures input***);

vii. Wherein the local area of textures is filtered utilizing a filter including a plurality of elements (***seventh page, right-side column, second and third paragraphs; please note that a convolution operation is a filter operation***);

viii. Wherein the input includes a local area of textures used to sample a texture map to generate a modified local area of textures (***second and third pages, section 2. Computational Setting; and third page, figure 1***).

e. Burden appears to teach:

i. Wherein the determining whether the solution has converged includes calculating errors and concluding that the solution has converged based on the calculation of the errors (*page 403, Jacobi Iterative Algorithm 7.1, step 4,if ||x – XO|| < TOL then OUTPUT(x1, . . . xn); STOP.; where ||x-XO|| calculates errors x-XO and takes the norm, see pages 384 – 392, and especially page 393, exercise 2 which defines a norm that sums the vector components*).

f. The motivation to use the art of Rumpf with the art of Press would have been the benefits recited in Rumpf that the presented strategy opens a wide area of numerical applications for hardware acceleration (*first page, Abstract, first paragraph*), and turns a graphics card into an ultrafast vector coprocessor (*first page, Abstract, first paragraph*), which would have been recognized by the ordinary artisan as benefits that allow faster processing.

g. The motivation to use the art of Burden with the art of Press would have been the benefit recited in Burden that iterative techniques are efficient in terms of computational time and computer storage for large systems that have a high percentage of zero entries which arises frequently in numerical solution of partial differential equations (pages 400 – 401, last paragraph of page 400 continued on page 401).

h. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Rumpf and the art of Burden with the art of Press to produce the claimed invention..

==

i. Regarding **claim 11**:

i. Claim 11 is taught as a subset of limitations as described in claim 10 above.

==

10. **Claims 26, 28 and 30 - 31** are rejected under 35 U.S.C. 103(a) as being unpatentable over Press (Press, William H.; Flannery, Brian P.; Teukolsky, Saul A.; Vetterling, William T.; "Numerical Recipes in C", 1988, Cambridge University Press) in view of Rumpf (Martin Rumpf et al.; "Using Graphics Cards for Quantized FEM Computations", September 3 – 5 2001, Proceedings of the IASTED International

Conference on Visualization, Imaging and Image Processing), further in view of Burden (Richard L. Burden and J. Douglas Faires, "Numerical Analysis", fourth edition, 1989, PWS-Kent Publishing Company, pages 383 - 393, 400 - 403 and 605 - 643).

a. Regarding **claim 26**:

b. Press appears to teach:

i. Processing input (**pages 673-676, section 17.5 Relaxation Methods for Boundary Value Problems; it would have been obvious that input is required to solve a partial differential equation, especially given the statement that an initial distribution relaxes to an equilibrium distribution on page 673**);

ii. Processing input to generate a solution to partial differential equations (**pages 673-676, section 17.5 Relaxation Methods for Boundary Value Problems**);

iii. Wherein the processing further includes determining whether the solution has converged (**page 674, Relaxation Methods for Boundary Value Problems; first paragraph, section that starts with "Thus the algorithm consists . . .", sentence, "This procedure is then iterated until convergence."**);

c. Press does not specifically teach:

i. ***A hardware graphics pipeline for*** processing input to generate a solution to partial differential equations ***wherein the solution to the partial differential equation is generated utilizing the hardware graphics pipeline for enhancing graphics processing operations performed by the hardware graphics pipeline***;

ii. ***Wherein the graphics processing operation performed by the hardware graphics pipeline is enhanced by determining a location of surfaces or objects for rendering purposes utilizing the solution to the partial differential equation generated utilizing the hardware graphics pipeline***;

iii. Wherein the input includes a local area of textures used to sample a texture map to generate a modified local area of textures;

iv. ***Wherein the determining whether the solution has converged includes calculating errors and concluding that the solution has converged based on the calculation of the errors***.

d. Rumpf appears to teach:

i. A hardware graphics pipeline for processing input to generate a solution to partial differential equations wherein the solution to the partial differential equation is generated utilizing the hardware graphics pipeline for enhancing graphics processing operations performed by the hardware graphics pipeline (***third page, section 3.1 Vector Representation, first paragraph; and seventh page, section 6. Linear Heat Equation, first paragraph***);

ii. Wherein the graphics processing operation performed by the hardware graphics pipeline is enhanced by determining a location of surfaces or objects for rendering purposes utilizing the solution to the partial differential equation generated utilizing the hardware graphics pipeline (***ninth page, figure 3, displays surfaces and objects rendered by utilizing the solution to a partial differential equation utilizing a hardware graphics pipeline***);

iii. Wherein the input includes a local area of textures used to sample a texture map to generate a modified local area of textures (***second and third pages, section 2. Computational Setting; and third page, figure 1***).

e. Burden appears to teach:

i. Wherein the determining whether the solution has converged includes calculating errors and concluding that the solution has converged based on the calculation of the errors (***page 403, Jacobi Iterative Algorithm 7.1, step 4,if ||x – XO|| < TOL then OUTPUT(x1, . . . xn); STOP.; where ||x-XO|| calculates errors x-XO and takes the norm, see pages 384 – 392, and especially page 393, exercise 2 which defines a norm that sums the vector components***).

f. The motivation to use the art of Rumpf with the art of Press would have been the benefits recited in Rumpf that the presented strategy opens a wide area of numerical applications for hardware acceleration (***first page, Abstract, first paragraph***), and turns a graphics card into an ultrafast vector coprocessor (***first page, Abstract, first paragraph***), which would have been recognized by the ordinary artisan as benefits that allow faster processing.

g. The motivation to use the art of Burden with the art of Press would have been the benefit recited in Burden that iterative techniques are efficient in terms of computational time and computer storage for large systems that have a high percentage of zero entries which arises

frequently in numerical solution of partial differential equations (***pages 400 - 401, last paragraph of page 400 continued on page 401***).

h. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Rumpf and the art of Burden with the art of Press to produce the claimed invention.

==

i. Regarding **claim 28**:

j. Press appears to teach:

i. Receiving boundary conditions (**pages 673-676, section 17.5 Relaxation Methods for Boundary Value Problems; it would have been obvious that boundary conditions are required to solve a partial differential equation, especially since the title of the section recites Boundary Value problems**);

ii. Computing the solution to generate the solution to the partial differential equations involving the boundary conditions (**pages 673-676, section 17.5 Relaxation Methods for Boundary Value Problems**);

iii. Determining whether the solution has converged (**page 674, first paragraph, subsection that starts with "Thus the algorithm . . .", sentence, "This procedure is then iterated until convergence."**);

iv. If the solution has not converged, repeating the computing and determining (**page 674, first paragraph, subsection that starts with "Thus the algorithm . . .", sentence, "This procedure is then iterated until convergence."**);

v. wherein the solution to the partial differential equation is generated (**pages 673-676, section 17.5 Relaxation Methods for Boundary Value Problems**);

k. Press does not specifically teach:

i. Computing the solution to the partial differential equations involving the boundary conditions ***at least some of the computing done in the hardware graphics pipeline***;

ii. ***Generating output utilizing the hardware graphics pipeline for display***;

iii. wherein the solution to the partial differential equation is generated ***utilizing the hardware graphics pipeline for enhancing graphics processing operations performed by the hardware graphics pipeline***;

iv. ***Wherein the graphics processing operations performed by the hardware graphics pipeline are enhanced by determining a location of surfaces or objects for rendering purposes utilizing the solution to the partial differential equation generated utilizing the hardware graphics pipeline***;

v. ***Wherein the input includes a local area of textures used to sample a texture map to generate a modified local area of textures***;

vi. ***Wherein the determining whether the solution has converged includes calculating errors and concluding that the solution has converged based on the calculation of the errors***.

l. Rumpf appears to teach:

i. Receiving input in the hardware graphics pipeline (***third page, figure 1***);

ii. Computing the solution to the partial differential equations at least some of the computing done in the hardware graphics pipeline wherein the solution to the partial differential equation is generated utilizing the hardware graphics pipeline for enhancing graphics processing operations performed by the hardware graphics pipeline (***third page, section 3.1 Vector Representation, first paragraph; and seventh page, section 6. Linear Heat Equation, first paragraph***);

iii. Generating output utilizing the hardware graphics pipeline for display (***ninth page, figure 3***);

iv. Wherein the graphics processing operations performed by the hardware graphics pipeline are enhanced by determining a location of surfaces or objects for rendering purposes utilizing the solution to the partial differential equation generated utilizing the hardware graphics pipeline (***ninth page, figure 3, displays surfaces and objects rendered by utilizing the solution to a partial differential equation utilizing a hardware graphics pipeline***);

v. Wherein the input includes a local area of textures used to sample a texture map to generate a modified local area of textures (***second and third pages, section 2. Computational Setting; and third page, figure 1***).

m. Burden appears to teach:

i. Wherein the determining whether the solution has converged includes calculating errors and concluding that the solution has converged based on the calculation of the errors (*page 403, Jacobi Iterative Algorithm 7.1, step 4,if | |x - XO| | < TOL then OUTPUT(x1,... xn); STOP.; where | |x-XO| | calculates errors x-XO and takes the norm, see pages 384 - 392, and especially page 393, exercise 2 which defines a norm that sums the vector components*).

n. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Rumpf and the art of Burden with the art of Press to produce the claimed invention.

==

o. Regarding **claim 30**:

p. Press appears to teach:

i. Receiving a first input (**pages 673-676, section 17.5 Relaxation Methods for Boundary Value Problems; it would have been obvious that input is required to solve a partial differential equation, especially given the statement that an initial distribution relaxes to an equilibrium distribution on page 673**);

ii. Processing the first input to generate a solution to a partial differential equation (**pages 673-676, section 17.5 Relaxation Methods for Boundary Value Problems**);

iii. wherein the solution to the partial differential equation is generated (**pages 673-676, section 17.5 Relaxation Methods for Boundary Value Problems**);

iv. Wherein the processing further includes determining whether the solution has converged (**page 674, Relaxation Methods for Boundary Value Problems; first paragraph, section that starts with "Thus the algorithm consists . . .", sentence, "This procedure is then iterated until convergence."**);

q. Press does not specifically teach:

i. Receiving a first input *into a hardware graphics pipeline*;

ii. Processing the first input to generate a solution to a partial differential equation *utilizing the hardware graphics pipeline*;

iii. Receiving a second input into the hardware graphics pipeline;

iv. Rendering the 3D graphics image utilizing the hardware graphics pipeline for display, wherein the rendering utilizes the second input and the result of the processing of the first input;

v. wherein the solution to the partial differential equation is ***generated utilizing the hardware graphics pipeline for enhancing graphics processing operations performed by the hardware graphics pipeline***;

vi. ***Wherein the graphics processing operations performed by the hardware graphics pipeline are enhanced by determining a location of surfaces or objects for rendering purposes utilizing the solution to the partial differential equation generated utilizing the hardware graphics pipeline***;

vii. ***Wherein the input includes a local area of textures used to sample a texture map to generate a modified local area of textures***;

viii. ***Wherein the determining whether the solution has converged includes calculating errors and concluding that the solution has converged based on the calculation of the errors***.

r. Rumpf appears to teach:

i. Receiving a first input into a hardware graphics pipeline (***third page, figure 1***);

ii. Processing the first input to generate a solution to a partial differential equation utilizing the hardware graphics pipeline (***eighth page, left-side column, second paragraph; please note that an initial noisy image is input***);

iii. Receiving a second input into the hardware graphics pipeline (***eighth page, left-side column, second paragraph; please note that a contrast enhancing function is input***);

iv. Rendering the 3D graphics image utilizing the hardware graphics pipeline for display, wherein the rendering utilizes the second input and the result of the processing of the first input (***ninth page, figure 3***);

v. wherein the solution to the partial differential equation is generated utilizing the hardware graphics pipeline for enhancing graphics processing operations performed by the hardware graphics pipeline (***seventh, eighth and ninth pages, section 7 Anisotropic Diffusion in Image Processing***);

vi. Wherein the graphics processing operations performed by the hardware graphics pipeline are enhanced by determining a location of surfaces or objects for rendering

purposes utilizing the solution to the partial differential equation generated utilizing the hardware graphics pipeline (***ninth page, figure 3, displays surfaces and objects rendered by utilizing the solution to a partial differential equation utilizing a hardware graphics pipeline***);

vii. Wherein the input includes a local area of textures used to sample a texture map to generate a modified local area of textures (***second and third pages, section 2. Computational Setting; and third page, figure 1***).

s. Burden appears to teach:

i. Wherein the determining whether the solution has converged includes calculating errors and concluding that the solution has converged based on the calculation of the errors (***page 403, Jacobi Iterative Algorithm 7.1, step 4,if ||x – XO|| < TOL then OUTPUT(x1, . . . xn); STOP.; where ||x-XO|| calculates errors x-XO and takes the norm, see pages 384 – 392, and especially page 393, exercise 2 which defines a norm that sums the vector components***).

t. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Rumpf and the art of Burden with the art of Press to produce the claimed invention.

==

u. Regarding **claim 31**:

v. Press appears to teach:

i. The first input comprises boundary conditions (**pages 673-676, section 17.5 Relaxation Methods for Boundary Value Problems; it would have been obvious that boundary conditions are required to solve a partial differential equation, especially since the title of the section recites Boundary Value problems**);

ii. The determining whether the solution has converged (**<u>page 674, first paragraph, subsection that starts with "Thus the algorithm . . .", sentence, "This procedure is then iterated until convergence."</u>**);

iii. If the solution has not converged, repeating the computing and determining (**<u>page 674, first paragraph, subsection that starts with "Thus the algorithm . . .", sentence, "This procedure is then iterated until convergence."</u>**);

==

11. **Claim 29** is rejected under 35 U.S.C. 103(a) as being unpatentable over Press (Press, William H.; Flannery, Brian P.; Teukolsky, Saul A.; Vetterling, William T.; "Numerical Recipes in C", 1988, Cambridge University Press), further in view of Burden (Richard L. Burden and J. Douglas Faires, "Numerical Analysis", fourth edition, 1989, PWS-Kent Publishing Company, pages 383 - 393, 400 - 403 and 605 - 643) further in view of Rumpf (Martin Rumpf et al.; "Using Graphics Cards for Quantized FEM Computations", September 3 – 5 2001, Proceedings of the IASTED International Conference on Visualization, Imaging and Image Processing).

a. Regarding **claim 29**:

b. Press appears to teach:

i. Receiving boundary conditions (**<u>pages 673-676, section 17.5 Relaxation Methods for Boundary Value Problems; it would have been obvious that boundary conditions are required to solve a partial differential equation, especially since the title of the section recites Boundary Value problems</u>**);

ii. computing the solution to the partial differential equation utilizing a relaxation operation involving the boundary conditions (**<u>pages 673-676, section 17.5 Relaxation Methods for Boundary Value Problems</u>**);

iii. determining whether the solution has converged (**<u>page 674, first paragraph, subsection that starts with "Thus the algorithm . . .", sentence, "This procedure is then iterated until convergence."</u>**);

iv. If the solution has not converged, repeating the computing and determining (**page 674, first paragraph, subsection that starts with "Thus the algorithm . . .", sentence, "This procedure is then iterated until convergence."**);

v. if the solution has converged, incrementing a time value (**page 658, second paragraph, sentence that starts, "To solve equation (17.2.8) . . ."**); and

vi. repeating the foregoing operations using the incremented time value (**page 658, second paragraph, sentence that starts, "To solve equation (17.2.8) . . ."**).

vii. wherein the solution to the partial differential equation is generated conditions (**pages 673-676, section 17.5 Relaxation Methods for Boundary Value Problems**);

c. Press does not specifically teach:

i. Receiving boundary conditions ***in the form of at least one of geometry and textures***;

ii. computing the solution to the partial differential equation utilizing a relaxation operation involving the boundary conditions ***at least some of the computing done in the hardware graphics pipeline***;

iii. determining whether the solution has converged by:

(1) calculating the errors,

(2) summing the errors, and

iv. concluding that the solution has converged if the sum of errors is less than a predetermined amount;

v. ***Generating output utilizing the hardware graphics pipeline for display***;

vi. wherein the solution to the partial differential equation is generated ***utilizing the hardware graphics pipeline for enhancing graphics processing operations performed by the hardware graphics pipeline***;

vii. ***Wherein the graphics processing operations performed by the hardware graphics pipeline are enhanced by determining a location of surfaces or objects for rendering purposes utilizing the solution to the partial differential equation generated utilizing the hardware graphics pipeline***;

viii. ***Wherein the input includes a local area of textures used to sample a texture map to generate a modified local area of textures***.

d. Burden appears to teach:

i. determining whether the solution has converged by:

(1) calculating the errors (***page 403, Jacobi Iterative Algorithm 7.1, step 4,if ||x - XO|| < TOL then OUTPUT(x1, . . . xn); STOP.; where ||x-XO|| calculates errors x-XO and takes the norm, see pages 384 – 392, and especially page 393, exercise 2 which defines a norm that sums the vector components***),

(2) summing the errors (***page 403, Jacobi Iterative Algorithm 7.1, step 4,if ||x - XO|| < TOL then OUTPUT(x1, . . . xn); STOP.; where ||x-XO|| calculates errors x-XO and takes the norm, see pages 384 – 392, and especially page 393, exercise 2 which defines a norm that sums the vector components***), and

ii. concluding that the solution has converged if the sum of errors is less than a predetermined amount (***page 403, Jacobi Iterative Algorithm 7.1, step 4,if ||x - XO|| < TOL then OUTPUT(x1, . . . xn); STOP.; where TOL is a predetermined amount and where ||x-XO|| calculates errors x-XO and takes the norm, see pages 384 – 392, and especially page 393, exercise 2 which defines a norm that sums the vector components***);

e. Rumpf appears to teach:

i. Receiving boundary conditions in the form of at least one of geometry and textures (***third page, figure 1***);

ii. wherein the solution to the partial differential equation is generated utilizing the hardware graphics pipeline for enhancing graphics processing operations performed by the hardware graphics pipeline (***seventh page, section 6. Linear Heat Equation, first paragraph***);

iii. Wherein the graphics processing operations performed by the hardware graphics pipeline are enhanced by determining a location of surfaces or objects for rendering purposes utilizing the solution to the partial differential equation generated utilizing the hardware graphics pipeline (***ninth page, figure 3, displays surfaces and objects rendered by utilizing the solution to a partial differential equation utilizing a hardware graphics pipeline***).

iv. Generating output utilizing the hardware graphics pipeline for display (***ninth page, figure 3***);

v. Wherein the input includes a local area of textures used to sample a texture map to generate a modified local area of textures (***second and third pages, section 2. Computational Setting; and third page, figure 1***).

f. The motivation to use the art of Burden with the art of Press would have been the benefit recited in Burden that iterative techniques are efficient in terms of computational time and computer storage for large systems that have a high percentage of zero entries which arises frequently in numerical solution of partial differential equations (***pages 400 – 401, last paragraph of page 400 continued on page 401***).

g. The motivation to use the art of Rumpf with the art of Press would have been the benefits recited in Rumpf that the presented strategy opens a wide area of numerical applications for hardware acceleration (***first page, Abstract, first paragraph***), and turns a graphics card into an ultrafast vector coprocessor (***first page, Abstract, first paragraph***), which would have been recognized by the ordinary artisan as benefits that allow faster processing.

h. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Burden and the art of Rumpf with the art of Press to produce the claimed invention.

==

12. **Examiner's Note:** Examiner has cited particular columns and line numbers in the references applied to the claims above for the convenience of the applicant. Although the specified citations are representative of the teachings of the art and are applied to specific limitations within the individual claim, other passages and figures may apply as well. It is respectfully requested from the Applicant in preparing responses, to fully consider the references in their entirety as potentially teaching all or part of the claimed invention, as well as the context of the passage as taught by the prior art or disclosed by the Examiner. **The entire body of all references are considered as being recited to teach the claimed invention**.

Conclusion

13. The prior art made of record and not relied upon is relevant to the Applicant's disclosure:

a. David Kincaid and Ward Cheney, "Numerical Analysis", 1991, Wadsworth Inc., pages 161 - 164 in chapter 4 Solving Systems of Linear Equations, teaches knowledge of the ordinary artisan regarding norms, especially the l_1 norm on page 162, equation 5, which sums the components of a vector.

b. U. Diewald, T. Preusser, M. Rumpf, R. Strzodka, "Diffusion models and their accelerated solution in image and surface processing", 2001, Acta Mathematica Universitatis Comenianae, Volume LXX, issue 1, pages 15 - 31, teaches knowledge of the ordinary artisan regarding using a graphics pipeline to solve a partial differential equation, including that the graphics processor executes commands in memory (page 26, section 5, first paragraph).

c. J.L. Bell and G.S. Patterson Jr., "Data organization in large numerical computations", The Journal of Supercomputing, Volume 1, Number 1, pages 105 - 136, teaches knowledge of the ordinary artisan for selecting a numerical method for a partial differential equation (page 130, last paragraph, and page 132, figure 13).

14. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Russ Guill whose telephone number is 571-272-7955. The examiner can normally be reached on Monday - Friday 10:00 AM - 6:30 PM.

15. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Paul Rodriguez can be reached on 571-272-3753. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300. Any inquiry of a general nature or relating to the status of this application should be directed to the TC2100 Group Receptionist: 571-272-2100.

16. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Russ Guill
Examiner
Art Unit 2123

RG

PAUL RODRIGUEZ
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100